Visual object detection system

Abstract:-

This study proposes a deep neural network architecture for the phone location detection model. Initially multiple images have been trained using VGG16 Architecture , a phone is considered to be detected correctly on a test image if the final object coordinates of the test image is within a radius of 0.05 (normalized distance) centered on the phone. Perfect detection performance is not the main goal of this test. For this prototype, your algorithm is expected to detect a phone correctly on 4 out of the 8 test images and to detect at least 70% correctly on the provided labeled dataset. If you do not have enough time, please focus on a submission with clean, well structured code, rather than on the perfect performance.

Input data details:-

Parameters	
Images	120
Number of epochs	200
Optimizer	Adam
Learning rate	1e - 4
batch size	16
Layers	5

Bounding box regression model using VGG net is used for object detection

```
bboxHead = Dense(128, activation="relu")(flatten)
bboxHead = Dense(64, activation="relu")(bboxHead)
bboxHead = Dense(32, activation="relu")(bboxHead)
bboxHead = Dense(16, activation="relu")(bboxHead)
bboxHead = Dense(8, activation="relu")(bboxHead)
bboxHead = Dense(2, activation="sigmoid")(bboxHead)
```

Train_phone_finder.py	This file will take the folder path which contains labels and images
Find_phone.py	It will take input image to be tested

Conclusion:-

The model worked with 70% accuracy with a test image of 90:10 split. This python script will be responsible for:

1. Loading our phone image training data from disk (i.e., both class labels and center coordinates)
2. Loading VGG16 (pre-trained on ImageNet), removing the fully-connected classification layer head from the network, and inserting our bounding box regression layer head
3. Fine-tuning the regression layer head on our training data